

Method for managing storage space in a memory with data groups and reference data inscribed on either side of free space in memory zone

Publication number: FR2790321 (A1)

Publication date: 2000-09-01

Inventor(s): ORLANDO WILLIAM

Applicant(s): ST MICROELECTRONICS SA [FR]

Classification:

- **international:** G06F17/30; G07F7/10; G06F17/30; G07F7/10; (IPC1-7); G06F12/02; G06K19/07

- **European:** G06F17/30G3; G07F7/10D10M2

Application number: FR19990002367 19990225

Priority number(s): FR19990002367 19990225

Also published as:

FR2790321 (B1)

Cited documents:

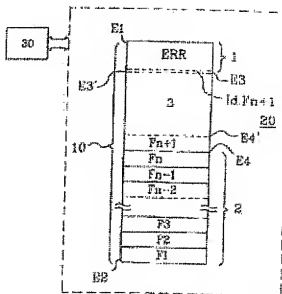
FR2635886 (A1)

WO9400822 (A1)

FR2752072 (A1)

Abstract of FR 2790321 (A1)

Method for managing memory (20) storage space accessible for writing into. Data groups and reference data is inscribed on either side of free space (3) in memory zone. Free space (3) is filled from each opposite end (E4, E3) by data groups and their associated reference information as and when data groups are written into memory. Data groups correspond to files, indexes, catalogue or sub-indexes or sub-catalogues. Independent claim for device for inscribing into memory.



Data supplied from the esp@cenet database — Worldwide

(19) RÉPUBLIQUE FRANÇAISE
**INSTITUT NATIONAL
 DE LA PROPRIÉTÉ INDUSTRIELLE**
 PARIS

(11) N° de publication : **2 790 321**
 (à n'utiliser que pour les
 commandes de reproduction)
 (21) N° d'enregistrement national : **99 02367**

(51) Int Cl⁷ : G 06 F 12/02, G 06 K 19/07

(12)

DEMANDE DE BREVET D'INVENTION

A1

(22) Date de dépôt : 25.02.99.

(30) Priorité :

(43) Date de mise à la disposition du public de la
 demande : 01.09.00 Bulletin 00/35.

(56) Liste des documents cités dans le rapport de
 recherche préliminaire : *Se reporter à la fin du
 présent fascicule*

(60) Références à d'autres documents nationaux
 apparentés :

(71) Demandeur(s) : **STMICROELECTRONICS SA**
Société anonyme — FR.

(72) Inventeur(s) : **ORLANDO WILLIAM.**

(73) Titulaire(s) :

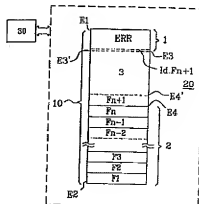
(74) Mandataire(s) : **CABINET BALLOT SCHMIT.**

(54) **PROCEDE DE GESTION DE L'ESPACE DE STOCKAGE D'UNE MEMOIRE, MEMOIRE ET DISPOSITIF
 D'INSCRIPTION EN MEMOIRE GERES SELON LEDIT PROCEDE.**

(57) Le procédé gère l'espace de stockage d'une mémoire (20) accessible en écriture. Il est destiné à enregistrer dans une zone mémoire (10), d'une part, des groupes de données et, d'autre part, des informations de repérage dans la zone mémoire de ces groupes de données. Selon ce procédé, pour l'enregistrement d'un groupe de données, on inscrit le groupe de données et les informations de repérage de celui-ci respectivement de part et d'autre d'un espace libre (3) de la zone mémoire.

Le procédé est adapté à l'enregistrement de fichiers, de répertoires et de sous-répertoires.

L'invention concerne également une mémoire, par exemple une carte à puce, organisée selon ce procédé. Elle concerne également un dispositif d'inscription de données fonctionnant selon ce procédé.



FR 2 790 321 - A1



2790321

1

**Procédé de gestion de l'espace de stockage d'une
mémoire, mémoire et dispositif d'inscription en mémoire
gérés selon ledit procédé**

La présente invention concerne un procédé de
gestion de l'espace de stockage d'une mémoire,
notamment d'une mémoire accessible en écriture et qui
comporte des informations d'identification de son
contenu. Elle concerne aussi une mémoire dont l'espace
de stockage est géré selon ce procédé, ainsi qu'un
dispositif d'inscription de données destiné à coopérer
avec cette mémoire selon ledit procédé.

Bien que particulièrement avantageuse pour des
mémoires de faible capacité, telles que les cartes à
puce, l'invention peut être exploitée avec tout type de
mémoire dont le contenu n'est pas figé, qu'il s'agisse
de mémoires du type semiconducteur, magnétique,
optique, magnéto-optique ou autre.

De manière classique, une mémoire est organisée
pour stocker des informations sous forme de blocs
distincts de données qui peuvent être assimilés à
différents fichiers.

Un fichier est un ensemble de données
indissociables vis-à-vis d'un système qui les
exploitent. Chaque fichier enregistré dans une mémoire
porte un nom unique par lequel il est identifié. Ce
nom peut contenir une partie propre à l'identification
du fichier, et éventuellement une partie qui décrit le
type d'informations contenues dans le fichier, par
exemple s'il s'agit d'un module d'un programme, d'un
texte, de données relatives à une application
particulière, etc.

2790321

2

Chaque fichier stocké dans une mémoire dépend directement d'un répertoire. Un répertoire est un contenant qui reçoit un ou plusieurs fichier(s). Il permet, par exemple, de réunir des fichiers d'un même type ou des fichiers liés à une même application. Chaque répertoire d'une mémoire comporte un nom spécifique par lequel il est identifié et qui, en général, rappelle un aspect commun des fichiers qu'il comporte. Des fichiers peuvent être ajoutés à un répertoire ou y être supprimés, si bien que la taille d'un répertoire, c'est-à-dire l'espace qu'il occupe dans une mémoire, est variable.

Un répertoire peut être rattaché à un autre répertoire. Dans ce cas, il porte la désignation de sous-répertoire. Il est aussi possible d'avoir un sous-répertoire qui dépend lui-même d'un autre sous-répertoire. Ce dernier peut aussi être rattaché à un sous-répertoire, et ainsi de suite, selon une concaténation de sous-répertoires.

L'organisation d'une mémoire vis-à-vis des répertoires et des sous-répertoires suit une structure arborescente. L'accès à un fichier nécessite de suivre un chemin spécifique de la structure arborescente, qui peut passer par un ou plusieurs sous-répertoire(s).

La structure arborescente des répertoires est modifiable. Il est notamment possible de créer et de supprimer des répertoires et des sous-répertoires. Il est également possible de modifier les liens de rattachement entre différents répertoires et sous-répertoires.

Un exemple de structure arborescente des fichiers d'une mémoire sera décrit brièvement par référence à la figure 1.

De manière classique, la mémoire comporte un répertoire de base, appelé répertoire "racine" RR à la

2790321

3

tête du tronc principal TP de l'arborescence. Le répertoire racine RR est le contenant global de tous les répertoires et fichiers de la mémoire. Il constitue le point de départ de tous les chemins d'accès au fichiers de la mémoire. Autrement dit, l'emplacement d'un fichier dans l'arborescence doit être donné par rapport au fichier racine RR.

Dans l'exemple, le répertoire racine RR comporte quatre fichiers F1-F4 qui lui sont directement rattachés par le tronc principal TP. De ce fait, on accède directement à ces fichiers F1-F4 depuis le répertoire racine par le tronc principal TP.

Le répertoire racine RR comporte également un premier et un deuxième répertoires R1 et R2 qui sont rattachés directement au tronc principal TP par des branches respectives b1 et b2.

Le premier répertoire R1 contient cinq fichiers F5-F9, qui lui sont directement rattachés. Pour accéder à l'un de ces cinq fichiers à partir du répertoire racine RR, il est nécessaire d'accéder d'abord au premier répertoire R1 (ce que l'on appellera par la suite "entrer dans le répertoire") suivant la branche b1. Ainsi, il est nécessaire de savoir que ces fichiers appartiennent au répertoire R1 et que celui-ci est relié directement au répertoire racine RR.

De même, le deuxième répertoire R2 contient trois fichiers F10-F12. Pour avoir accès à l'un de ces fichiers, il d'abord nécessaire d'entrer dans le répertoire R2 suivant la branche b2, comme pour le cas précédent.

Le premier répertoire R1 comporte également un sous-répertoire SR1 qui lui est directement rattaché par la branche b3. Ce sous-répertoire SR1 contient quatre fichiers F13-F16 qui lui sont directement rattachés. Pour accéder à l'un de ces fichiers F13-

2790321

4

F16, il est nécessaire d'entrer dans le sous-répertoire SR1, ce qui demande d'entrer d'abord dans le répertoire R1. Ainsi, pour accéder aux fichiers F13-F16, il est nécessaire de savoir d'une part que ces derniers sont reliés directement au sous-répertoire SR1, et d'autre part que ce sous-répertoire dépend du répertoire R1, rattaché au tronc principal TP.

Le sous-répertoire SR1 contient lui-même un sous-répertoire SR1-1, qui lui est relié par la branche b4. (Dans ce qui suit, on maintient la désignation de "sous-répertoire" pour tout répertoire dépendant d'un autre répertoire (à part le répertoire racine RR) ou d'un autre sous-répertoire, quel que soit le niveau d'arborescence de ce dernier.)

Le sous-répertoire SR1-1 contient trois fichiers F17-F19. Pour accéder à l'un de ces fichiers, il est nécessaire de parcourir le chemin qui permet d'entrer successivement dans le répertoire R1 par la branche b1, puis dans le sous-répertoire SR1 par la branche b3, et enfin dans le sous-répertoire SR1-1.

Afin de permettre de trouver l'adresse en mémoire d'un fichier, le répertoire racine contient une table d'arborescence des fichiers. Cette table associe chaque fichier avec son chemin d'accès à partir du répertoire racine, répertoriant notamment les éventuels répertoires et sous-répertoires auxquels il est associé. La table rassemble ainsi des informations telles que représentées schématiquement dans l'exemple de la figure 1. Elle indique en outre l'adresse spécifique des fichiers dans la mémoire, par exemple en désignant pour chaque fichier l'adresse ou le secteur qui contient la première donnée du fichier.

La table d'arborescence des fichiers est contenue dans une portion prédéterminée de la mémoire, de sorte qu'un dispositif de lecture et/ou d'inscription (appelé

2790321

5

dispositif d'accès en mémoire ci-après) peut y accéder directement. Une fois le dispositif positionné sur la table, il lui est possible de retrouver immédiatement l'adresse en mémoire d'un fichier donné en suivant son arborescence.

5

La partie active de la mémoire est ainsi décomposée en deux portions de stockage : une première portion pour contenir la table d'arborescence des fichiers et une seconde portion pour contenir l'ensemble des fichiers.

10

La figure 2 représente schématiquement une organisation classique de ces deux portions dans l'espace de stockage 10 d'une mémoire.

La taille de la première portion 1 est déterminée à un stade initial de la mise en service de la mémoire et est normalement figée ensuite. Classiquement, cette portion occupe une série d'adresses située à une extrémité E1 de la partie de stockage de la mémoire, par exemple le début.

15

La deuxième portion 2 constitue le complément de l'espace de stockage 10 de la mémoire. Les fichiers y sont stockés en chaîne suivant l'ordre chronologique de leur inscription, et à partir de l'extrémité E3 de cette deuxième portion 2 qui jouxte la première portion 1. L'espace libre 3 de la deuxième portion 2 se réduit donc par cette extrémité E3 au fur et à mesure que des nouveaux fichiers y sont stockés. La mémoire est saturée lorsque toutes les adresses libres de la deuxième portion 2 sont occupées, ou lorsque l'espace libre n'est plus suffisant pour y inscrire un fichier complet.

20

25

30

Du fait que le nombre de fichiers, de répertoires et de sous-répertoires est évolutif, il est nécessaire de réserver à la table d'arborescence des fichiers dans la première portion 1 une capacité telle qu'elle puisse

35

2790321

6

répertorier le nombre maximum de fichiers, de répertoires et de sous-répertoires. Ce nombre maximum est gouverné non seulement par la capacité de la mémoire, mais aussi par la constitution des fichiers, notamment la taille du plus petit fichier envisageable, et de l'organisation interne de la mémoire. Ces deux derniers paramètres peuvent donner un nombre théorique de fichiers pouvant être stockés par répertoire qui est bien au-delà du plus grand nombre de fichiers qui y seront réellement stockés à un moment donné. Cependant, du fait que ce dernier nombre est généralement imprévisible lors de la configuration de la table, on confère à celle-ci une dimension adaptée à ce nombre théorique afin de parer à tous les cas de figure possibles. En particulier, il est nécessaire de prévoir pour la table des emplacements libres séparément pour chaque répertoire. On remarque ici que la répartition des fichiers par répertoire étant inconnu lors de la configuration de la table, le nombre total de ces emplacements libres peut être supérieur au nombre global de fichiers pouvant être stocké en mémoire.

Il résulte de cette démarche que la première portion 1 de la mémoire, dédiée à la table d'arborescence des fichiers, est surdimensionnée au détriment de l'espace 2 pouvant être dédié à la deuxième portion de stockage de fichiers.

Cet inconvénient est particulièrement pénalisant lorsque la capacité de la mémoire est restreinte. En effet, plus la mémoire est petite, plus la perte de capacité exploitable en raison du surdimensionnement de la table est importante par rapport à la capacité totale de la mémoire.

Il existe d'autres organisations de mémoire qui ne souffrent pas autant de cet inconvénient. Selon ces

2790321

7

organisations, la table d'arborescence des fichiers est remplacée par une simple liste de fichiers et de leur adresses respectives dans la mémoire.

5 La partie de la mémoire dédiée à la liste doit demeurer suffisante pour répertorier le nombre maximum de fichiers possible. Cependant, on ne considère ici que le nombre global de fichiers de la mémoire, et non pas le nombre de fichiers pouvant être contenu dans chaque répertoire, comme dans le cas d'une organisation
10 en table d'arborescence des fichiers. De la sorte, la liste des fichiers utilise l'espace mémoire plus efficacement qu'une table d'arborescence des fichiers.

En général, la liste n'est pas organisée d'une manière qui facilite la recherche d'un fichier. Par
15 exemple, chaque nouveau fichier est tout simplement ajouté à la fin de la liste. Ainsi, lorsque le dispositif d'accès doit rechercher un fichier donné, il parcourt la liste de façon séquentielle depuis le début jusqu'à ce que ce fichier soit atteint. Cette
20 organisation est donc lente par rapport à celle basée sur une table d'arborescence des fichiers qui, comme expliqué plus haut, permet d'accéder directement à un fichier.

Au vu de ces problèmes de l'art antérieur, la
25 présente invention propose un procédé de gestion de l'espace de stockage d'une mémoire accessible en écriture, destinée à enregistrer dans une zone mémoire d'une part des groupes de données et d'autre part des informations de repérage dans la zone mémoire de ces
30 groupes de données, caractérisé en ce que l'on inscrit ledit groupe de données et les informations de repérage de celui-ci respectivement de part et d'autre d'un espace libre de la zone mémoire.

Ainsi, le procédé selon la présente invention se
35 démarque des approches classiquement utilisées selon

2790321

8

lesquelles on inscrit les groupes de données à partir du premier emplacement qui jouxte la partie de mémoire, définie d'avance, réservée à l'enregistrement des informations de repérage, c'est-à-dire sans jamais
5 laisser d'espace libre entre cette partie prédéfinie et les groupes de données.

Il en résulte qu'avec le procédé de l'invention il n'est plus nécessaire de définir d'avance la taille et la configuration de la partie de mémoire destinée à
10 stocker les informations de repérage. En effet, cette partie de mémoire peut se développer à tout moment en entamant l'espace libre précité de la zone mémoire, et ce jusqu'au remplissage complet de cette zone.

Cet espace libre peut se remplir par chaque extrémité opposée respectivement par les groupes de données et leurs informations de repérage associées, au fur et à mesure qu'on y inscrit des groupes de données. A tout moment, la partie de la zone mémoire dédiée aux informations de repérage n'occupe que l'espace qui lui
15 est nécessaire à ce moment. Sa taille évolue de façon dynamique exactement en fonction des besoins, et sa configuration suit parfaitement celle des groupes de données inscrits.

Qui plus est, la partie de la zone mémoire dédiée aux informations de repérage est utilisée de manière très efficace comparativement aux tables d'arborescence des fichiers classiques, puisqu'il n'est pas nécessaire d'y créer des espaces vides pour d'éventuels répertoires, ces espaces étant par ailleurs souvent
20 imbriqués dans des sections occupées et difficilement reconfigurables.

On remarque que le procédé conforme à l'invention est récursif puisque l'espace libre de la zone mémoire se situe toujours de part et d'autre l'espace dédié aux
35 informations de repérage des données et de l'espace

2790321

9

dédié au stockage de ces données. Autrement dit, l'espace libre a toujours une extrémité qui jouxte l'espace dédié aux informations de repérage et un extrémité opposée qui jouxte l'espace dédié au stockage de données. Ces extrémités évoluent chacune dans le sens du rétrécissement de l'espace libre au fur et à mesure que des nouvelles données sont enregistrées et répertoriées. La position de l'une au moins de ces extrémités peut être utilisé comme paramètre variable dans la détermination de l'adresse de stockage d'un groupe de données.

La notion d'extrémité dans le contexte d'une zone mémoire se rapporte au fait qu'une mémoire est organisée en une succession d'adresses commençant par une première adresse et se terminant par une dernière adresse. Ces première et dernière adresses sont considérées, sur la plan de la topographie de la mémoire, comme étant situées à deux extrémités respectives de la zone mémoire. Il ne s'agit là que d'une représentation symbolique, et l'homme du métier comprendra bien que ces extrémités n'ont pas à correspondre forcément à des extrémités matérialisées dans la structure réelle d'une mémoire pour que l'invention puisse être mise en oeuvre.

Les groupes de données précités peuvent correspondre à des fichiers, des répertoires ou à des sous-répertoires.

La mémoire peut comporter une unique zone mémoire ayant un seul espace libre. Dans ce cas, la mémoire se remplit par avancement depuis sa première adresse et à rebours depuis sa dernière adresse, respectivement avec l'un ou l'autre des groupes de données et des informations de repérage de celles-ci.

Le procédé de l'invention permet également des organisations plus complexes de l'espace total de la

2790321

10

mémoire. Cet espace peut être découpé en plusieurs sous-espaces indépendants, pour chacun desquels l'enregistrement d'un groupe de données s'effectue en inscrivant ledit groupe de données et les informations de repérage celui-ci respectivement de part et d'autre de la partie libre de ce sous-espace. Autrement dit, chaque sous-espace est géré de la même manière que l'espace mémoire précité en référence au procédé caractéristique de l'invention.

A titre d'exemple, on peut allouer un sous-espace du type précité pour un répertoire ou sous-répertoire respectif de la mémoire.

Les sous-espaces peuvent être créés à un stade initial de la préparation de la mémoire. Il est possible de reconfigurer les sous-espaces à tout moment de l'exploitation de la mémoire.

Selon une mise en oeuvre avantageuse de l'invention, les informations de repérage des données sont structurées selon un format uniforme pour chaque groupe de données stocké. A titre d'exemple, on utilise le même format pour les informations de repérage, que les données stockées concernent des fichiers, des répertoires ou des sous-répertoires.

De préférence, les informations de repérage d'un groupe de données stockées comprennent l'une au moins des informations suivantes concernant le groupe: i) son adresse en mémoire, ii) son type, iii) sa taille.

L'invention concerne également une mémoire accessible en écriture, par exemple une carte à puce, destinée à enregistrer dans une zone mémoire d'une part des groupes de données et d'autre part des informations de repérage dans la zone mémoire de ces groupes de données, caractérisée en ce qu'elle enregistre le groupe de données et les informations de repérage

2790321

11

celui-ci respectivement de part et d'autre d'un espace libre de la zone mémoire.

Enfin, l'invention concerne également un dispositif d'inscription de groupes de données dans la mémoire précitée caractérisé en ce que le dispositif inscrit ledit groupe de données et les informations de repérage de celui-ci respectivement de part et d'autre d'un espace libre de la zone mémoire.

D'autres avantages et caractéristiques de l'invention apparaîtront plus clairement à la lectures des modes de réalisation préférés, donnés purement à titre d'exemples non limitatifs, en référence aux dessins annexés dans lesquels :

- la figure 1, déjà présentée, montre un exemple de structure arborescente de fichiers, de répertoires et de sous-répertoires ;

- la figure 2, déjà présentée, est un schéma de principe d'une répartition classique d'une mémoire en portions dédiées respectivement au stockage de fichiers et au stockage d'une table d'arborescence des fichiers ;

- la figure 3 montre une arborescence de fichiers enregistrés dans une mémoire selon l'exemple 1 de mise en oeuvre du procédé de l'invention ;

- la figure 4 représente la topographie de l'utilisation de la zone mémoire conformément à l'invention pour enregistrer l'arborescence de l'exemple 1 ;

- la figure 5 est un schéma visant à expliquer la détermination de l'adresse du premier fichier enregistré dans la zone mémoire ;

- la figure 6 montre une arborescence de fichiers enregistrés dans une mémoire selon l'exemple 2 de mise en oeuvre du procédé de l'invention ;

2790321

12

- la figure 7 représente la topographie de l'utilisation de la zone mémoire conformément à l'invention pour enregistrer l'arborescence de l'exemple 2 ;

5 - la figure 8 montre une arborescence de fichiers enregistrés dans une mémoire selon l'exemple 3 de mise en oeuvre du procédé de l'invention ; et

10 - la figure 9 montre une arborescence de fichiers enregistrés dans une mémoire selon l'exemple 3 de mise en oeuvre du procédé de l'invention.

15 Le procédé l'invention sera maintenant décrit par le biais de trois exemples de mise en oeuvre, chacun correspondant à un cas d'enregistrement de groupes de données sous forme fichiers suivant une arborescence donnée.

20 Chaque exemple est illustré par une première figure qui indique schématiquement l'arborescence des fichiers enregistrés en mémoire, une deuxième figure qui indique la topographie correspondante de l'utilisation de la zone mémoire conformément à l'invention, et un tableau qui présente la configuration des données stockées dans cette zone mémoire.

25 La mémoire dont il sera fait référence est accessible en enregistrement et est adaptée pour stocker des groupes de données sous forme de fichiers, de répertoires ou de sous-répertoires qui sont répertoriés dans un répertoire racine.

30 Les exemples peuvent s'appliquer notamment à des mémoires semiconducteur, du type mémoire vive (RAM en anglais) utilisées dans des matériels informatiques ou dans des cartes à puce servant à la téléphonie, aux transactions bancaires, au commerce électronique etc. On peut aussi considérer les exemples dans le cadre de
35 mémoires magnétiques (disques durs, ensembles lecteur

2790321

13

et disquette), optiques ou magnéto-optiques sous diverses formes. On remarque que l'invention est particulièrement intéressante dans le cas de mémoires destinées aux cartes à puce et autres applications dans lesquelles la capacité de stockage de données est restreinte.

Ces mémoires étant bien connues de l'homme du métier, leur description détaillée au niveau de leur construction et de leur mode opératoire sera omise par souci de concision.

Le terme "zone mémoire" désigne spécifiquement la partie active de la mémoire dédiée au stockage des fichiers, répertoires et sous-répertoires ainsi que les informations de repérage de ces derniers. De la sorte, on ne tient pas compte dans la description qui suit d'éventuelles parties actives de la mémoire pouvant servir à stocker des données relatives à d'autres applications, qu'elles soient internes ou externes.

La zone mémoire peut être considérée comme un espace dans lequel les différentes données peuvent être casées. Cet espace est divisé en segments élémentaires qui sont individuellement repérés par une adresse. Selon le type de mémoire utilisé, les segments peuvent correspondre à une cellule de stockage d'un seul élément binaire (bit). Ceci est généralement le cas avec des mémoires du type semiconducteur. Un segment peut aussi correspondre à un emplacement pour plusieurs éléments binaires, par exemple un groupe de mots binaires selon un format préétabli. Ceci est généralement le cas avec des mémoires magnétiques ou optiques, où les segments correspondent par exemple à des secteurs de mémoire.

L'attribution d'une partie spécifique de la zone mémoire à un groupe de données est gérée par un contrôleur de mémoire. Son rôle consiste notamment à

2790321

14

déterminer la taille des données présentées en mémoire, c'est-à-dire l'espace qu'elles occuperont dans la zone mémoire, puis de déterminer si celle-ci dispose d'un espace libre suffisant pour stocker ces données.

5 Ensuite, le contrôleur attribue au groupe de données un emplacement dans la zone mémoire 10 en associant à ce groupe des informations concernant sa taille, ses liens avec d'autres données et son emplacement. L'emplacement peut être enregistré en termes de l'adresse du premier

10 segment d'une suite de segments qui stockent les données du groupe. Ces informations sont formatées et transmises au répertoire racine RR sous forme d'en-tête.

Dans la description qui suit, on utilise les termes et repères ayant servi pour les figures 1 et 2 dans la mesure où ils peuvent être considérés comme désignant des éléments analogues.

15

Exemple 1 : enregistrement d'une succession de fichiers dépendant tous directement du répertoire racine.

20

Cet exemple correspond à l'un des plus simples cas possibles d'arborescence de fichiers. Ainsi que le montre la figure 3, les données enregistrées correspondent à un nombre n de fichiers $F_1, F_2, F_3, \dots, F_{n-2}, F_{n-1}, F_n$, tous répertoriés ensemble et dépendant directement du tronc commun TC d'un répertoire racine RR. Comme il sera décrit plus en

25 détail par référence au tableau 1, l'en-tête ERR du répertoire racine RR intègre la table d'arborescence des fichiers avec les informations d'identification et l'adresse en mémoire de chacun d'entre eux

30

On suppose que la quantité de données que comporte l'ensemble des n fichiers $F_1, F_2, F_3, \dots, F_{n-2}, F_{n-1}, F_n$ et l'en-tête ERR du répertoire racine RR est inférieure à la capacité de stockage de la zone

35

2790321

15

mémoire 10, de sorte que celle-ci peut recevoir des nouveaux fichiers.

La figure 4 montre la topographie de l'enregistrement des fichiers F1, F2, F3, ..., Fn-2, Fn-1, Fn et de l'en-tête ERR du répertoire racine RR dans la zone mémoire 10 d'une mémoire 20. Dans cet exemple, la zone mémoire 10 occupe l'intégralité de l'espace de stockage de données de la mémoire 20.

La mémoire 20 est associée à un dispositif 30 d'accès en mémoire, qui peut être par exemple un lecteur de carte à puce capable aussi d'inscrire des données dans celle-ci.

Conformément à la présente invention, l'en-tête ERR du répertoire racine RR et l'ensemble des fichiers F1, F2, F3, ..., Fn-2, Fn-1, Fn sont localisés à des parties respectives 1 et 2 qui sont séparées par une partie libre 3 de la zone mémoire 10. Cette partie libre 3 est disponible tant pour agrandir l'en-tête ERR du répertoire racine RR que pour stocker des nouveaux fichiers.

La partie 1 de la zone mémoire 10 occupée par l'en-tête ERR du répertoire racine RR est située entre une première extrémité E1 de cette zone et l'extrémité E3 de la partie libre 3 la plus proche de cette extrémité E1. L'extrémité E3 précitée est désignée ci-après l'extrémité libre côté répertoire racine. De même, la partie de la zone mémoire 10 occupée par les fichiers F1, F2, F3, ..., Fn-2, Fn-1, Fn est située entre une deuxième extrémité E2 de la zone mémoire 10 et l'extrémité E4 de la partie libre 3 la plus proche de cette extrémité E2. L'extrémité E4 précitée est désignée ci-après l'extrémité libre côté fichiers.

Lorsqu'il est nécessaire d'agrandir l'en-tête ERR du répertoire racine RR, la partie libre 3 est attaquée par l'extrémité libre côté répertoire racine E3. On

2790321

16

comprendra que cette extrémité E3 se confond avec la première extrémité E1 de la zone mémoire 10 lorsque la mémoire est vierge. De même, lorsqu'on stocke les données qui forment le contenu d'un nouveau fichier, la partie libre 3 est attaquée par l'extrémité libre côté fichiers E4. Cette extrémité E4 se confond avec la deuxième extrémité E2 lorsque la mémoire est vierge.

Les éléments d'identification des données de la figure 4 seront détaillés par report au tableau 1.

L'en-tête ERR du répertoire racine RR comporte une première entrée permettant sa propre identification (entrée 1). Cette entrée spécifie le type de la racine, c'est-à-dire le fait qu'il s'agit d'un répertoire, son adresse et sa taille totale. Ces informations sont inscrites à un stade initial, avant tout stockage de fichiers. La taille dont il est ici question est la taille de la zone mémoire 10 entière, puisque le répertoire racine rassemble tous les fichiers de tous les répertoires et sous-répertoires.

Ensuite, l'en-tête ERR indique la première adresse libre de la racine, la taille L de l'espace de stockage libre dans la racine, ainsi que le nombre m de fichiers et sous-répertoires qu'il comporte (entrée 2). Ces paramètres L et m varient en fonction des données enregistrées et de l'arborescence de ces fichiers ; ils sont mis à jour à chaque modification de l'état du répertoire racine RR.

2790321

17

Tableau 1 : structure en mémoire de l'arborescence de l'exemple 1

5	Entrée	Contenu de l'en-tête du répertoire racine (ERR)
	1	Identificateur de la Racine, type = répertoire, adresse, taille totale
10	2	Première adresse libre pour la Racine, taille libre dans la Racine (=L), Nombre m de fichiers et sous-répertoires = 0
	3	Identificateur de fichier 1, type = fichier, adresse A1, taille T1
15	4	Identificateur de fichier 2, type = fichier, adresse A2, taille T2
	5	Identificateur de fichier 3, type = fichier, adresse A3, taille T3
	—	—
	—	—
	—	—
20	n+2	Identificateur de fichier n, type = fichier, adresse An, taille Tn
	Espace disponible du répertoire Racine	
25	Données du répertoire racine	
	Données fichier n	
30	Données fichier 3	
	Données fichier 2	
	Données fichier 1	

2790321

18

Dans l'exemple, n fichiers sont stockés dans le répertoire racine RR, mais celui-ci ne contient aucun autre répertoire ou sous-répertoire. Les valeurs de ces paramètres sont donc n. L'espace de stockage L libre dans la racine correspond à la capacité de la partie libre 3 de la zone mémoire 10.

Ensuite l'en-tête ERR enregistre les informations relatives à chacun de fichiers F1, F2, F3, ..., Fn-2, Fn-1, Fn contenus dans la zone mémoire 10.

Ces informations sont enregistrées successivement au fur et à mesure qu'un nouveau fichier est stocké dans la zone mémoire 10. Si on suppose que l'ordre d'enregistrement des fichiers suit numériquement la succession des nombres 1 à n de leur désignation, la première information d'identification concerne le fichier F1. Cette information correspond donc à la l'entrée 3 du tableau 1. Elle est enregistrée dans la zone mémoire 10 immédiatement après l'entrée 2 précitée. On comprend en effet que juste avant son enregistrement, l'extrémité E3 côté en-tête répertoire racine de la partie libre 3 de la zone mémoire 10 s'étendait jusqu'à la première adresse libre après la dernière adresse occupée par les informations de l'entrée 2 précitée.

Les données qui constituent le fichier F1 sont stockées de manière à aboutir à l'autre extrémité E2 de la zone mémoire 10. En effet, lors de l'enregistrement de ce premier fichier F1, la partie libre 3 de la zone mémoire 10 s'étendait jusqu'à cette extrémité E2.

Lors de l'opération d'enregistrement du fichier F1 et de son en-tête, l'entrée 2 du tableau 1 est modifiée pour mettre à jour le paramètre f désignant le nombre fichiers de $f = 0$ à $f = 1$. Durant cette opération, on inscrit à l'en-tête ERR les informations

2790321

19

qui figurent à l'entrée 3 du tableau 1, c'est-à-dire le type de données (fichier), son adresse et sa taille.

Pour l'enregistrement de ces informations, on commence par créer l'espace nécessaire pour cet enregistrement dans l'en-tête ERR du répertoire racine RR. L'en-tête s'accroît en conséquence sur l'espace libre 3 de la zone mémoire 10. On désigne alors la première adresse libre de l'espace libre 3 comme étant celle qui succède à la dernière adresse occupée par les informations d'enregistrement du fichier F1 dans l'en-tête ERR. Cette première adresse libre correspond à l'emplacement de l'extrémité libre E3 côté répertoire racine, et est désignée adresse E3.

On enregistre ensuite la taille T1 du fichier F1, c'est-à-dire le nombre d'adresses successives qu'il occupe dans la zone mémoire 10. La taille d'un fichier est une donnée a priori en ce sens qu'elle est indiquée dans le fichier lui-même à un emplacement prédéterminé de ce dernier.

Enfin, on enregistre dans l'en-tête ERR l'adresse du fichier F1 dans la zone mémoire (entrée 3 du tableau 1). Dans l'exemple, cette adresse correspond à l'adresse à laquelle est enregistrée la première donnée du fichier F1. Du fait que le fichier F1 est enregistré à partir de l'extrémité de l'espace libre 3 à l'opposée de celle à laquelle se situe l'en-tête, conformément à la présente invention, cette première adresse est déterminée de la façon suivante, expliquée par référence à la figure 5.

On commence avec comme point de référence l'adresse E3 précitée, c'est-à-dire la première adresse libre de l'espace libre 3 après l'inscription de l'entrée 3 du tableau 1. On ajoute à cette adresse la taille L de l'espace libre avant l'enregistrement des données qui constituent le fichier F1, pour obtenir

2790321

20

l'adresse E3+L. La taille L de l'espace libre est ici exprimée comme le nombre d'adresses libres successives entre l'extrémité libre E3 côté en-tête (qui correspond donc à la première adresse libre précitée) et l'extrémité libre E4 côté fichiers (qui correspond, pour le premier fichier F1, à la dernière adresse de la zone mémoire 10 située à l'extrémité E2). Comme il sera vérifié plus loin, on obtient par cette addition systématiquement l'adresse qui correspond à dernière adresse libre de la partie libre 3, quel que soit l'état d'occupation de la mémoire.

De la sorte, l'adresse qui correspond à la première adresse du fichier F1 est obtenue en soustrayant de l'adresse E3+L la taille du fichier F1. Ainsi, l'adresse A1 du premier fichier F1 est égale à la première adresse libre de la partie libre 3 plus la taille L de la mémoire libre et moins la taille T1 du fichier F1, soit :

$$A1 = E3+L-T1 \dots (1).$$

Le procédé d'enregistrement de fichiers et de leur en-tête conformément à l'invention est récursif en ce sens que l'enregistrement de chaque nouveau fichier engendre une évolution des paramètres de calcul d'adresse pour l'enregistrement du fichier suivant, de façon à permettre d'appliquer de nouveau la même procédure.

Ce caractère récursif peut être mieux compris si l'on considère l'état de la zone mémoire 10 de la figure 4, dans le cas où on enregistre un nouveau fichier, désigné F_{n+1} et représenté en pointillées.

Avant l'enregistrement du fichier F_{n+1}, la dernière adresse disponible de la partie libre 3 de la zone mémoire 10 est celle, située à l'extrémité E4, qui précède la première adresse du fichier F_n. La première adresse libre de la partie libre 3, située à

2790321

21

l'extrémité donnée par la variable E3, succède la dernière adresse occupée par l'en-tête ERR contenant les informations concernant le fichier Fn (entrée n+2 du tableau 1), ajoutée des adresses occupées par l'inscription du fichier Fn+1. On obtient l'adresse An+1 du fichier Fn+1 en appliquant la formule (1) supra exactement de la même manière que pour le fichier F1, celle-ci étant donnée dans ce cas par :

An+1 = E3+L-Tn+1 ; où Tn+1 est la taille du fichier Fn+1.

Exemple 2 : enregistrement de fichiers dépendant de deux répertoires rattachés au répertoire racine.

Dans cet exemple, le répertoire racine RR comporte, au moment considéré, un premier et un deuxième répertoires R1 et R2, contenant respectivement des fichiers F1-1 à F1-k et F2-1 à F2-m, où k et m sont des nombres entiers. Le répertoire racine RR ne contient aucun autre fichier, qu'il s'agisse de fichiers rattachés directement ou indirectement par le biais de répertoires ou de sous-répertoires. L'arborescence du répertoire racine est représentée à la figure 6.

L'organisation correspondante de la zone mémoire 10 du répertoire racine RR est représentée à la figure 7.

Cette organisation suit le même principe que pour le cas de l'exemple 1. Les premières adresses de la zone mémoire 10, à l'extrémité E1 contiennent l'en-tête ERR du répertoire racine RR. La dernière adresse de l'en-tête ERR est à l'extrémité E3. La partie 3 de la zone mémoire entre cette extrémité E3 et celle E2 opposée à l'extrémité E1 constitue donc l'espace de stockage des répertoires R1 et R2.

La zone mémoire 10 précitée est partagée en trois sous-espaces contigus : un premier sous-espace 10a

2790321

22

dédié au stockage d'éventuels nouveaux répertoires ou sous-répertoires et de leurs fichiers, un deuxième sous-espace 10b dédié au stockage du répertoire R1 et de ses fichiers, et un troisième sous-espace 10c dédié au stockage du répertoire R2 et de ses fichiers. La répartition des trois sous-espaces 10a-10c est déterminé à un stade initial de la configuration de la zone mémoire 10. Elle peut être cependant modifiée pour une optimisation de l'occupation de la zone mémoire 10 selon des techniques en elles-mêmes connues.

Chacun de ces sous-espaces 10a-10c est traité comme un espace indépendant en ce qui concerne la mise en oeuvre du procédé de gestion conforme à l'invention.

Ainsi, le premier sous-espace 10a, qui commence à la première extrémité E1, est partagé entre la partie 1a occupée par l'en-tête ERR du répertoire racine RR et la partie 3a dédiée au stockage d'éventuels nouveaux répertoires, sous-répertoires ou fichiers. L'occupation de cette partie 3a s'effectue à partir de son extrémité E2 à l'opposé de l'extrémité E1 à laquelle débute l'en-tête ERR. Les inscriptions des éventuels fichiers s'opèrent de la même manière que pour l'exemple 1, notamment en ce qui concerne l'application de la formule (1) pour la détermination de la première adresse.

Le deuxième sous-espace 10b succède directement à la suite du premier sous-espace 10a. Il comporte une première partie 1b occupée par l'en-tête ER1 du répertoire R1 et d'une deuxième partie dédiée au stockage des fichiers. Cette deuxième partie comporte une partie libre 3b et une partie 2b occupée par les fichiers F1-1 à F1-k. Lorsqu'il est nécessaire de stocker un nouveau fichier dans le répertoire R1, on enregistre son identificateur dans l'en-tête ER1, en déterminant son adresse selon la formule (1). Dans ce

2790321

23

cas, on tient compte dans la formule que la première adresse libre est celle qui succède à son identificateur et la taille L de la mémoire libre est celle de la partie libre 3a du sous-espace 10b avant le stockage du nouveau fichier.

Le troisième sous-espace 10c succède directement à la suite du deuxième sous-espace 10b et se termine à l'extrémité E2 de la zone mémoire 10. Son organisation et sa gestion sont strictement identiques à celles du deuxième sous-espace et ne seront par décrites à nouveau par souci de concision. On note seulement que pour le stockage d'un nouveau fichier, on tient compte dans la formule (1) que la première adresse libre est celle qui succède à son identificateur dans l'en-tête ER2 du deuxième répertoire R2 et que la taille L de la mémoire libre est celle de la partie libre 3c du sous-espace 10c avant le stockage du nouveau fichier.

La structure en mémoire de l'arborescence de l'exemple 2 est donnée dans le tableau 2.

On constate que l'en-tête ERR du répertoire racine a la même structure que celle de l'exemple 1, seules diffèrent d'une part les paramètres f et r de l'entrée 2 qui spécifient respectivement le nombre fichiers et de répertoires qui dépendent directement du répertoire racine, et d'autre part les identificateurs des répertoires 1 et 2, respectivement aux entrées 3 et 4.

2790321

24

Tableau 2 : structure en mémoire de l'arborescence de l'exemple 2

5	Entrée	Contenu de l'en-tête du répertoire racine (ERR)
	1	Identificateur de la Racine, type = répertoire, adresse, taille totale
	2	Première adresse libre pour la Racine, taille libre dans la Racine (=L), Nombre de fichiers et sous-répertoires = 2
10	3	Identificateur répertoire 1, type = répertoire, adresse, taille
	4	Identificateur répertoire 2, type = répertoire, adresse, taille
		Espace disponible du répertoire Racine
15		Contenu de l'en-tête du répertoire R1
	5	Première adresse libre pour répertoire 1, taille libre dans répertoire 1, nombre de fichiers et sous-répertoires = k (k fichiers et 0 sous-répertoires)
20	6	Identificateur de fichier 1-1, type = fichier, adresse A1-1, taille T1-1
	7	Identificateur de fichier 1-2, type = fichier, adresse A1-2, taille T1-2
	8	
25	k+5	Identificateur de fichier 1-k, type = fichier, adresse A1-k, taille T1-k
		Espace disponible du répertoire 1
30		Données du répertoire 1
		Données fichier 1-k
		Données fichier 1-1
		Contenu de l'en-tête du répertoire R2
	k+6	Première adresse libre pour répertoire 2, taille libre dans répertoire 2, nombre de fichiers et sous-répertoires = m (m fichiers et 0 sous-répertoires)
35	k+7	Identificateur de fichier 2-1, type = fichier, adresse A2-1, taille T2-1
	k+8	Identificateur de fichier 2-2, type = fichier, adresse A2-2, taille T2-2
40	k+m+5	Identificateur de fichier 2-k, type = fichier, adresse A2-k, taille T2-k
		Espace disponible du répertoire 2
45		Données du répertoire 2
		Données fichier 1-m
		Données fichier 1-1
50		

2790321

25

Les deux répertoires R1 et R2 ont rigoureusement la même structure. Cette structure est par ailleurs analogue à celle de l'en-tête ERR, hormis le fait que cette dernière comporte en outre une première ligne "identificateur racine". Aussi, les entrées qui figurent au tableau 2 seront compréhensibles à la lumière de la description des figures 6 et 7 ainsi qu'à l'explication déjà donnée du tableau 1 supra.

Exemple 3 : enregistrement d'un fichier rattaché directement au répertoire racine ainsi que de 2 répertoires et d'un sous-répertoire.

Dans cet exemple, les fichiers stockés dans le répertoire racine RR sont organisés selon l'arborescence représentée à la figure 8. Un premier et deuxième répertoires R1 et R2, ainsi qu'un fichier F3 sont directement rattachés au répertoire racine RR.

Le premier répertoire R1 comporte un fichier F1-1 et un sous-répertoire R1-2. Le sous répertoire R1-2 contient un seul fichier F1-2-1.

Le deuxième répertoire R2 contient un seul fichier F2-1 qui lui est directement rattaché.

L'organisation correspondante de la zone mémoire 10 du répertoire racine RR est représentée à la figure 9.

La zone mémoire 10 est partagée entre ses deux extrémités E1 et E2 en quatre sous-espaces successifs 10a à 10d chacun dédié à un répertoire ou sous-répertoire respectif comme suit :- sous-espace 10a : répertoire racine RR ; sous-espace 10b : deuxième répertoire R2 ; sous-espace 10c : premier répertoire R1 ; sous-espace 10 d : sous-répertoire R1-2.

Chaque sous-espace 10a-10d comporte une partie en-tête 1a-1d et une partie libre 3a à 3d pour l'enregistrement de nouveaux fichiers.

2790321

26

De même que pour l'exemple 2, chaque sous-espace 10a à 10d est géré séparément en ce qui concerne la détermination des adresses des fichiers qui y sont stockés. Autrement dit, chaque sous-espace est considéré comme une zone mémoire à part, ayant sa propre taille, une première adresse et une dernière adresse. Les premières adresses d'un sous-espace sont utilisées pour l'enregistrement de l'en-tête du répertoire qui lui est associé.

Lorsqu'un fichier doit être enregistré, on détermine d'abord le répertoire ou sous-répertoire auquel il est destiné. Conformément à l'invention, le sous-espace 10a à 10d correspondant est alors géré en sorte que l'identificateur du fichier et les données du fichier sont enregistrés respectivement de part et d'autre de l'espace libre 3a à 3d de ce sous-espace. On procède ici de la même manière que pour l'exemple 1 ou 2, notamment en ce qui concerne la détermination de l'adresse du fichier selon la formule (1) supra.

La structure en mémoire associée à l'arborescence de la figure 8 est donnée par le tableau 3.

2790321

27

Tableau 3 : structure en mémoire de l'arborescence de l'exemple 3
 Identificateur de la Racine, type = répertoire, adresse, taille totale

5	CONTENU DU REPERTOIRE RACINE
10	<hr/> première adresse libre pour Racine, taille libre dans Racine nombre de fichiers et sous-répertoires = 3 (2 répertoires, 1 fichier) Identificateur Rep 1, type = répertoire, adresse, taille Identificateur Rep 2, type = répertoire, adresse, taille Identificateur Fic3, type = fichier, adresse, taille <hr/>
15	ESPACE DISPONIBLE DU REPERTOIRE RACINE
	FIC 3 : Données fichier Fic 3
20	<hr/> <u>Répertoire 2 :</u> première adresse libre pour Rep 2, taille libre dans Rep 2 nombre de fichiers et sous-répertoires = 1 (1 fichier) Id Fic2.1, type = fichier, adresse, taille <hr/>
25	ESPACE DISPONIBLE DU REPERTOIRE Rep 2
	FIC 2.1 : Données fichier Fic 2.1
30	<hr/> <u>Répertoire 1 :</u> première adresse libre pour Rep1, taille libre dans Rep1 nombre de fichiers et sous-répertoires = 2 (1 fichier, 1 répertoire) Id Fic1.1, type = fichier, adresse, taille Id Rep 1.2, type = répertoire, adresse, taille <hr/>
35	ESPACE DISPONIBLE DU REPERTOIRE Rep 1
	<hr/> <u>Répertoire 1.2 :</u> première adresse libre pour Rep 1.2, taille libre dans Rep 1.2 nombre de fichiers et sous-répertoires = 1 (1 fichier) Id Fic 1.2.1, type fichier, adresse, taille <hr/>
40	ESPACE DISPONIBLE DU REPERTOIRE Rep 1.2
	FIC 1.2 : Données fichier Fic 1.2.1
45	<hr/> FIC 1.1 : Données fichier Fic 1.1 <hr/>

2790321

28

On constate dans ce tableau l'identité de structure des identificateurs de fichier (Id.Fic.) et de répertoires (Id.Rep.). Les entrées qui figurent à chaque en-tête des quatre répertoires/sous-répertoire, à savoir le répertoire racine, les premiers et deuxième répertoires et le sous-répertoire 1-2 suivent le même format, analogue à celui des deux exemples précédents. Ces entrées, compréhensibles en elles-mêmes dans le contexte l'arborescence décrit par référence aux figures 8 et 9, ne seront donc pas décrits à nouveau par souci de concision.

Les portions du contenu du répertoire racine dédiés aux différents répertoires, sous-répertoire et fichiers sont indiquées dans les colonnes à la droite de la colonne principale.

On remarque que dans cet exemple, lorsqu'un répertoire contient un sous-répertoire, comme il en est le cas avec le premier répertoire R1 qui contient le sous-répertoire R1-2, les données de leurs fichiers respectifs sont stockées ensemble dans l'espace libre du sous-répertoire.

Dans les exemples venant d'être décrits, la procédure d'enregistrement (i.e. de création) d'un fichier dans un répertoire ou sous-répertoire donné (dénommé ci-après "répertoire parent") se résume comme suit.

Le dispositif d'accès en mémoire 30 (figure 4) fournit à la mémoire 20 l'identificateur et la taille du fichier devant être enregistré. Ces informations sont inscrites dans l'en-tête du répertoire parent et le nombre de fichiers (et éventuellement de sous-répertoires) est incrémenté en créant une nouvelle

2790321

29

ligne d'en-tête à l'extrémité de l'espace qui jouxte l'en-tête.

Cette ligne d'en-tête contient l'identificateur du fichier, son type (fichier), l'adresse à laquelle il est stocké, et sa taille. En application de la formule (1) supra, l'adresse à laquelle est stocké le fichier est calculée comme étant la première adresse libre du sous-espace 10a à 10d du répertoire parent plus la taille de la mémoire libre (i.e. l'espace libre 3a à 3d) - la taille du fichier. Le dispositif d'accès pourra alors inscrire le contenu du fichier dans l'espace libre à cette adresse.

En vertu du caractère récursif du procédé, la première adresse libre et la taille disponible sont mises à jour en tenant compte des modifications de paramètres. Pour chaque répertoire, on a donc la zone mémoire non encore utilisée qui se réduit par une extrémité de la taille d'une ligne d'en tête et par l'autre extrémité de la taille affectée au nouveau fichier.

Conformément à l'invention, il est également possible d'ajouter (i.e. créer) un nouveau répertoire ou sous-répertoire de manière analogue. La procédure dans ce cas est comme suit.

Le dispositif d'accès en mémoire 30 fournit à la mémoire 20 la taille maximale à affecter au répertoire/sous-répertoire ainsi que son identificateur, qui doit être différent des identificateurs des autres fichiers et sous-répertoires ayant le même père. En réponse, la mémoire 20 rajoute une ligne à l'en-tête du répertoire parent et le nombre de fichiers/répertoires contenus est ainsi incrémenté.

La ligne d'en-tête contient l'identificateur du répertoire, son type (répertoire), l'adresse à laquelle il est stocké et la taille qui lui est affectée. Cette

2790321

30

adresse est calculée de la même manière que pour l'adresse d'un nouveau fichier, conformément à la formule (1) supra, adaptée au cas d'espèce comme suit :

5 adresse nouveau répertoire/sous-répertoire =
première adresse libre dans le répertoire parent plus
la taille du sous-répertoire devant être créé moins la
taille du sous-répertoire.

10 La mémoire inscrit à cette adresse une en-tête décrivant le contenu du sous-répertoire et contenant les informations suivantes : première adresse libre, taille mémoire libre et nombre de fichiers et sous-répertoires contenus (ce nombre est égal à 0 au moment de la création).

15 La première adresse libre et la taille sont mises à jour en tenant compte des modifications.

20 L'accès à un fichier ou à un répertoire depuis le répertoire racine RR se fait en parcourant l'arborescence. L'accès à partir d'un niveau de l'arborescence à un fichier ou sous-répertoire de niveau immédiatement inférieur se fait en parcourant la table décrivant le contenu de ce répertoire. Les en-têtes ayant la même structure, et donc la même taille qu'ils décrivent un fichier ou un sous répertoire, les en-têtes conformément à l'invention sont donc
25 comparables à des tables d'arborescence des fichiers et sont donc gérés plus rapidement qu'une liste de fichiers et de répertoires utilisée dans l'art antérieur.

30 Bien entendu, les exemples de mode de réalisation de l'invention qui viennent d'être présentés ne sont nullement limitatifs. On comprendra notamment que les formats utilisés pour les identificateurs sont arbitraires et correspondent seulement à un protocole donné.

2790321

31

De même, lorsque l'espace mémoire 10 est divisé en sous-espaces, il n'est pas nécessaire que ces derniers se succèdent selon un ordre particulier, dès lors que leurs emplacements relatifs soient connus au niveau du contrôleur de mémoire.

5

2790321

32

R E V E N D I C A T I O N S

1. Procédé de gestion de l'espace de stockage d'une mémoire (20) accessible en écriture, destinée à enregistrer dans une zone mémoire (10), d'une part, des groupes de données et, d'autre part, des informations de repérage dans la zone mémoire de ces groupes de données, caractérisé en ce que l'on inscrit ledit groupe de données et les informations de repérage de celui-ci respectivement de part et d'autre d'un espace libre (3) de la zone mémoire.
2. Procédé selon la revendication 1, caractérisé en ce que l'espace libre (3) se remplit par chaque extrémité opposée (E4, E3) de celui-ci, respectivement par les groupes de données et leurs informations de repérage associées, au fur et à mesure qu'on y inscrit des groupes de données.
3. Procédé selon la revendication 1 ou 2, caractérisé en ce que les groupes de données correspondent à des fichiers, à des répertoires ou à des sous-répertoires.
4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que la mémoire (20) comporte une unique zone mémoire (10) ayant un seul espace libre (3).
5. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que la zone mémoire (10) est découpée en plusieurs sous-espaces indépendants (10a à 10d), pour chacun desquels un enregistrement s'effectue en inscrivant ledit groupe de données et les informations de repérage celui-ci

2790321

33

respectivement de part et d'autre de la partie libre de ce sous-espace.

6. Procédé selon la revendication 5, caractérisé en ce que qu'un sous-espace (10a à 10d) est alloué à un répertoire ou à un sous-répertoire respectif de la mémoire (20).

7. Procédé selon l'une quelconque des revendications 1 à 6, caractérisé en ce que les informations de repérage des données sont structurées selon un format uniforme pour chaque groupe de données stocké.

8. Procédé selon l'une quelconque des revendications 1 à 7, caractérisé en ce que les informations de repérage d'un groupe de données stockées comprennent l'une au moins des informations suivantes concernant le groupe: i) son adresse en mémoire, ii) son type, et iii) sa taille.

9. Procédé selon l'une quelconque des revendications 1 à 8, caractérisé en ce que l'adresse d'un groupe de données enregistré dans la zone mémoire (10) ou dans un sous-espace (10a à 10d) de celle-ci est déterminée comme étant la première adresse libre à l'extrémité (E3) de l'espace libre (3) à l'opposé de celle (E4) où sont stockés les groupes de données plus la taille de l'espace libre lors de l'enregistrement moins la taille du groupe de données à enregistrer.

10. Mémoire (20) accessible en écriture destinée à enregistrer dans une zone mémoire (10) d'une part des groupes de données et d'autre part des informations de repérage dans la zone mémoire de ces groupes de

2790321

34

données, caractérisée en ce qu'elle enregistre chaque
groupe de données en stockant le groupe de données et
les informations de repérage celui-ci respectivement de
part et d'autre d'un espace libre (3) de la zone
5 mémoire.

11. Mémoire (20) organisée selon le procédé de
l'une quelconque des revendications 1 à 9.

10 12. Carte à puce comprenant une mémoire (20)
selon la revendication 10 ou 11.

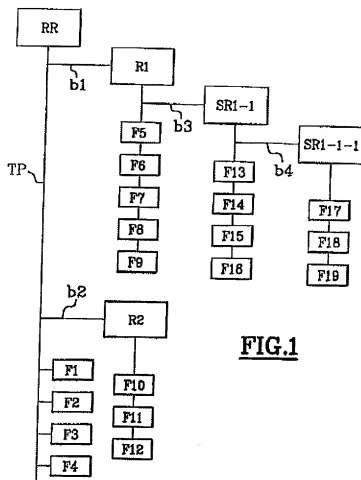
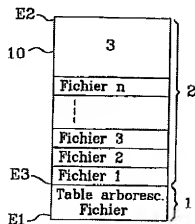
13. Dispositif (30) d'inscription de groupes de
données dans une mémoire (20) caractérisé en ce que,
15 pour chaque enregistrement de groupe de données, le
dispositif inscrit ledit groupe de données et des
informations de repérage celui-ci respectivement de
part et d'autre d'un espace libre (3) d'une zone
mémoire (10) de la mémoire.

20 14. Dispositif (30) d'inscription de groupes de
données dans une mémoire (20), caractérisé en ce qu'il
organise l'inscription des données conformément au
procédé de l'une quelconque des revendications 1 à 8.

25 15. Dispositif (30) selon la revendication 13 ou
14 destiné à fonctionner avec une mémoire (20) intégré
à une carte à puce.

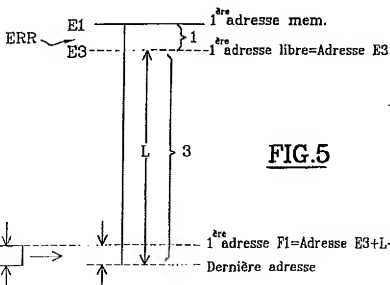
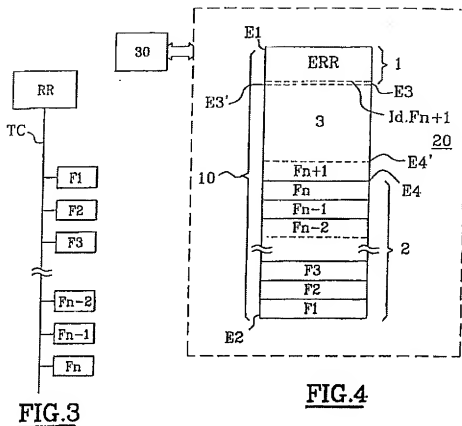
2790321

1/4

**FIG.1****FIG.2**

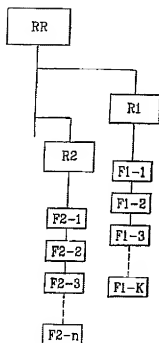
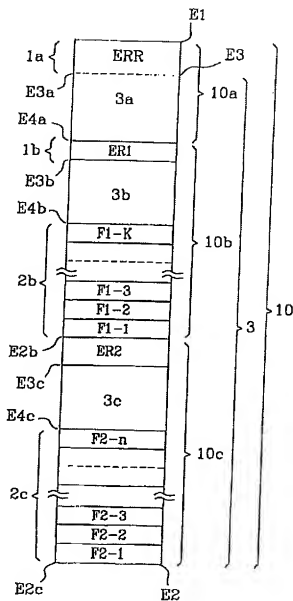
2790321

2/4



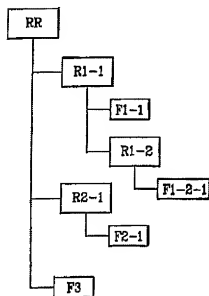
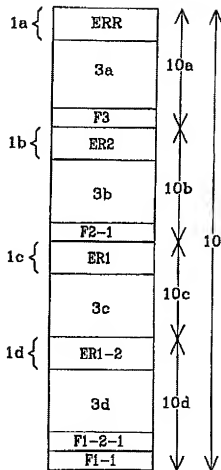
2790321

3/4

FIG.6FIG.7

2790321

4/4

**FIG.8****FIG.9**

REPUBLIQUE FRANÇAISE

2790321

INSTITUT NATIONAL

de la

PROPRIÉTÉ INDUSTRIELLE

RAPPORT DE RECHERCHE
PRELIMINAIREétabli sur la base des dernières revendications
déposées avant le commencement de la rechercheN° d'enregistrement
national

FA 573584

FR 9902367

DOCUMENTS CONSIDERES COMME PERTINENTS		Revendications concernées de la demande examinée
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	
X	FR 2 635 886 A (TOKYO SHIBAURA ELECTRIC CO) 2 mars 1990 (1990-03-02) * abrégé; figure 4 * * page 1, ligne 1 - page 3, ligne 19 *	1-3,5-15
A	WO 94 00822 A (GEMPLUS CARD INT ; SOURENIAN PAUL (FR)) 6 janvier 1994 (1994-01-06) * abrégé; figure 1 * * page 1, ligne 1 - page 3, ligne 19 *	1-15
A	FR 2 752 072 A (SOLAIC SA) 6 février 1998 (1998-02-06) * abrégé *	1-15
		DOMAINES TECHNIQUES RECHERCHES (Int.Cl.6)
		G06F
Date d'achèvement de la recherche		Examineur
4 novembre 1999		Katerbau, R
CATEGORIE DES DOCUMENTS CITES		
X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : pertinen à l'encontre d'au moins une revendication ou arrière-plan technologique général O : divulgation non-écrite P : document intercalaire T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant		

1
EPO FORM 1503 (8.92) (page 13)